Demo: A Demonstration of Striim A Streaming Integration and Intelligence Platform

Alok Pareek
Striim Inc., Palo Alto, USA
alok@striim.com

Bohan Zhang
Striim Inc., Palo Alto, USA
bohan@striim.com

Bhushan Khaladkar
Striim Inc., Palo Alto, USA
bhushan@striim.com

ABSTRACT

Today's data-driven applications need to process, analyze and act on real-time data as it arrives. The massive amount of data is continuously generated from multiple sources and arrives in a streaming fashion with high volume and high velocity, which makes it hard to process and analyze in real time. We introduce Striim, a distributed streaming platform that enables real-time integration and intelligence. Striim provides high-throughput, low-latency event processing. It can ingest streaming data from multiple sources, process data with SQL-like query language, analyze data with sophisticated machine learning models, write data into a variety of targets, and visualize data for real-time decision making. In this demonstration, we showcase Striim's ability to collect, integrate, process, analyze and visualize large streaming data in real time.

CCS CONCEPTS

• **Computer systems organization → Real-time systems**;

KEYWORDS

streaming, machine learning, data integration

ACM Reference format:
Alok Pareek, Bohan Zhang, and Bhushan Khaladkar. 2019. Demo: A Demonstration of Striim A Streaming Integration and Intelligence Platform. In *Proceedings of DEBS '19: The 13th ACM International Conference on Distributed and Event-based Systems, Darmstadt, Germany, June 24–28, 2019 (DEBS '19)*, 4 pages.
https://doi.org/10.1145/3328905.3332519

1 INTRODUCTION

Enterprises today are producing high-volume, high-velocity, high-variety real-time data continuously. To be data-driven, they will need to understand and respond to data as it arrives so that the data becomes valuable at the very instant it is born. In many real world use cases, one cannot gain value from analyzing historical, after-the-fact data with a batch-oriented offline process. Although there are a number of great machine learning tools for large dataset like Spark,Tensorflow, and Pytorch, they cannot assist real-time business decision in many cases. For example, a credit card fraud detection model needs to decide whether to approve a transaction immediately after a card holder swipes a credit card for a purchase. A malware detection model must stop a download process of a detected malicious file as soon as the download request from a victim is made. Many data-driven businesses need to process and analyze streaming data in real time for making immediate decisions.

An end-to-end real-time analysis over large streaming data has many challenges. First, it requires an efficient integration of real-time streaming data from multiple sources with high velocity, high throughput and variety. Take a popular bot detection system Shape security for example, it sees over 4 billion transactions per week, indicating that in each second more than 6000 transactions will be processed. Second, it needs to perform distributed data processing and transformation at scale. For instance, the process of collecting IoT sensor data may result in a large number of missing values and noises, putting a heavy duty on data filtering, data cleaning, and data transformation. Third, machine learning models should be efficient enough for real-time decision making, and can continuously evolve on the unbounded streaming data. A static model training and testing from historical data may not work well with newly generated data. Instead, a machine learning model needs to be updated periodically or continuously in an ideal way to guarantee a high predictive performance. Also, some time-consuming and resource-consuming models like deep neural network may not work well in the streaming scenario. Finally, operational systems demand human interpretable insights continuously extracted from machine learning operation but with minimal human intervention. Most of the existing data visualization tools are batch or event driven. Nonetheless, real-time visualization is required for immediacy of decision making. Although there exist distributed streaming platforms like Flink[2] and Kafka, one cannot build an end-to-end pipeline in a single system. e.g., Flink and Kafka do not support real-time visualization.

Given these observations, we introduce Striim [1], an advanced distributed streaming integration and intelligence platform. Striim meets all of the above non-trivial requirements for end-to-end real-time intelligence over big data streams. The platform can continuously (1) ingest and integrate massive amounts of data from multiple data sources, then (2) filter, enrich, aggregate, transform the streaming data with high throughput, (3) analyze the data using SQL-like language and adaptive machine learning models, and (4) visualize the real-time data and associated analyzed results, alert on issues or anomalies if necessary. Striim runs extremely fast by exploiting modern in-memory computing techniques. It also guarantees Exactly-Once-Processing semantics and fault tolerance.

Below we provide an overview of Striim in Section 2. We then describe the details of streaming integration (Section 3.1) and streaming intelligence (Section 3.2) in Striim. We conclude in Section 4 with our description of the demonstration.

DEBS '19, June 24–28, 2019, Darmstadt, Germany

ACM ISBN 978-1-4503-6794-3/19/06.
https://doi.org/10.1145/3328905.3332519

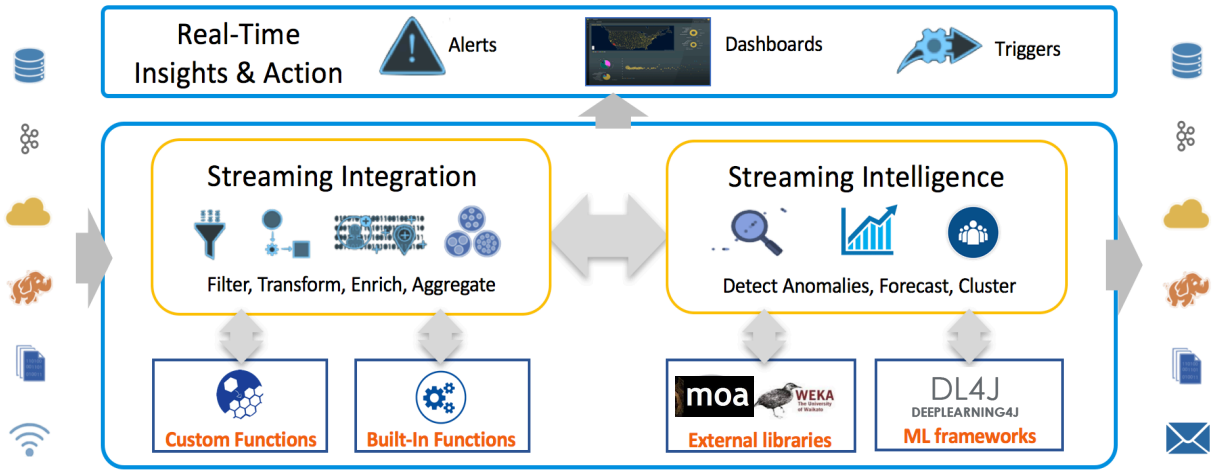

Figure 1: Striim Architecture – Striim continuously ingests real-time data from a wide variety of sources including databases, log files, IoT devices, message queues, and distributed file systems. Then it processes, transforms, filters, enriches, aggrates the streaming data with SQL-like language and custom functions. Next, Striim analyzes the real-time data with supervised and unsupervised machine learning models using built-in functions and external Java-based libraries like Weka, MOA and Deeplearning4J. Finally, it writes the data into a variety of targets, visualizes real-time data, sends alerts and triggers emails.

2 OVERVIEW

Striim is an end-to-end, enterprise-grade platform that moves, transforms and analyzes data with sub-second latency [6]. It combines nonintrusive, real-time change data capture capabilities with in-flight data processing and data visualization to deliver timely and enriched data to the enterprise. Striim provides an intuitive development experience with wizard-based user interface and speeds time-to-deployment with pre-built data pipelines. Striim uses a SQL-like language that is familiar to both business analysts and developers. Striim has a distributed scale-out architecture, and can continuously ingest massive data and then process and analyze with high-throughput, low latency by exploiting modern in-memory computing techniques. In Striim, it ingests data from a source, apply a transformation with a SQL-like query, and write the transformed data to a new stream (data pipe), which can be consumed by other transformation operators. Users can build a very complex data pipeline consisting of multiple streams and transformations. Striim has a cost-based optimizer to generate good query execution plan in its processing engine. Striim is fault-tolerant and guarantees Exactly-Once processing semantics.

Unlike existing tools which address either streaming integration or streaming analytics, Striim supports both in an integrated way as illustrated in Figure 1. Striim ingests real-time data from a wide variety of sources including databases, log files, IoT devices, message queues, for different data types such as JSON, XML, delimited, binary, free text, change records. For transactional databases, it uses nonintrusive change data capture (CDC). Striim runs continuous queries to filter, transform, aggregate, enrich, and analyze data-in-motion before delivering it to targets with sub-second latency.

Striim also offers advanced streaming analytics with predictive analytics capabilities to accurately discover time-sensitive insights. It provides a collection of machine learning models for supervised and unsupervised learning, especially for real-time prediction and anomaly detection. With built-in data visualization, one can monitor key business metrics in real time. Also, via real-time alerts and triggering workflows, it enables automated response to critical operational events. Its ease-of-use drives fast time-to-market and easy modification of analytical applications. It meets the strict security, reliability, and scalability requirements of business-critical solutions. Overall, Striim combines the power of real-time data capture, integration, transformation, analytics with SQL-like language, advanced analytics with machine learning, fault-tolerance and visualization in a single distributed streaming platform.

3 INTEGRATION AND INTELLIGENCE

Streaming integration and streaming intelligence are not independent. Many business-critical applications require both real-time integration and intelligence in a single pipeline. Take network traffic anomaly detection as an example, it may need to ingest data from multiple sources like on-premise Oracle databases, preprocess the data and fit it into machine learning models to detect anomalies, then write the results to targets Google Cloud's Spanner. Striim supports both integration and intelligence in a single platform.

3.1 Streaming Integration

Streaming Integration is an emerging paradigm in distributed event processing environments. High throughput Data Collection over the last few years has evolved from using ETL/ELT, CDC technologies to a Streaming Integration framework.

Data integration has been the cornerstone of the digital innovation for the last several decades, enabling the movement and processing of data across the enterprise to support data-driven decision making. In decades past, when businesses collected and shared data primarily for strategic decision making, batch-based ETL (Extract-Transform-Load) solutions served these organizations well. As consumers demand faster transaction processing, personalized experience, and self-service with up-to-date data access across multiple systems, a more efficient, continuous, real-time data integration approach is desired. In response, logical data replication with change data capture (CDC) capabilities emerged. To reduce high Extract (The E in ETL) costs, CDC moves only the change

data in real time, as opposed to all available data as a snapshot, and delivers data to downstream systems. These new technologies enabled businesses to create real-time replications of their databases to support master-slave workloads, higher throughput, and allow real-time operational decision making.

For many open-source databases like MySQL and PostgreSQL, CDC events can be captured using open-source libraries. However, there are no such tools for legacy databases like Oracle, which house most enterprise transactional data. Striim has a large number of built-in adapters that are able to capture transactional changes made in databases and output them in a streaming fashion, and then write the stream into target databases to achieve real-time collection and integration[7]. Supported databases include all legacy systems like Oracle, Microsoft SQL Server, DB2. It also supports a wide variety of additional data sources and targets, including file based like access logs, message based like Kafka, network based like TCP/UDP sockets, directory based like HDFS, etc. Different data formats like JSON, XML, CSV, free text are supported as well.

After ingesting data from multiple sources, real-time transformation, filtering, enrichment, aggregation can be applied to the data stream before writing to targets. Striim provides a wide range of SQL-like operators like select, join, where, order by, group by, etc. It also supports jumping window and sliding window based on either timestamp or event count. Users can easily integrate streaming data from multiple sources, and transform data on the fly in Striim.

3.2 Streaming Intelligence

To enable real-time business intelligence over large streaming data, machine learning models in Striim need to: (1) have low latency, (2) apply to unbounded and evolving data stream efficiently, and (3) integrate with the Striim platform seamlessly.

In Striim, one can use third-party libraries like Weka[4] and MOA (Massive Online Analysis), as well as Striim built-in linear regression models to build a real-time machine learning pipeline easily. To guarantee the low latency, Striim will not use time-consuming models like deep neural network which is more suitable to batch analysis with enough dataset and computing resource. Instead, It will choose more efficient models like random forest and gaussian process regression. Furthermore, Striim bounds the data size with a sliding window, trains machine learning models in memory, and updates models periodically. The update frequency can be controlled by simply defining the output interval in the sliding window. e.g., given a count-based sliding window with size 1000 and interval 10, the model will use previous 1000 data points as training data, and will be updated every 10 new data points come in. With efficient machine learning models, adaptive model training, in-memory computation, as well as bounded data size and update frequency, Striim can analyze large sets of data for real-time decision making.

To integrate with the current platform seamlessly, all of built-in machine learning models and third-party libraries in Striim are written in Java. There are a plethora of Java-based third-party libraries we can use, e.g., Weka is a collection of machine learning algorithms for data mining, MOA is an open source framework for data stream mining, and Deeplearning4j is a deep learning library. Since Striim is a Java-based system, it is hard and inefficient to use popular R library or Python library like scikit-learn in Striim. Although there are some interfaces (like JRI, Jython) to enable calling

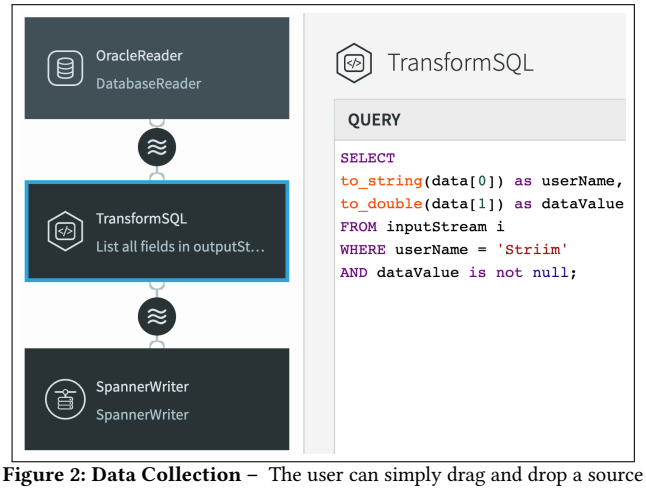

Figure 2: Data Collection – The user can simply drag and drop a source component (OracleReader), a SQL component (TransformSQL), and a target component (SpannerWriter) in Striim's UI to move data from Oracle to Spanner, and filter data with 'Striim' username and non null value on the fly. In the demonstration, we will invite audience to write their own queries.

R or Python functions in Java, they have some limitations. e.g., R is single-threaded internally so that two R functions cannot be called simultaneously. Also, using NumPy in Java is not well supported, which is a fundamental package for many other Python libraries.

A number of machine learning models are integrated with Striim seamlessly. Users can apply these models to their streaming data for data preprocessing, automated prediction, anomaly detection, etc. It is very simple and flexible to use these models in Striim. e.g., the following is to forecast the future value based on previous ones with random forest model.

```
SELECT forecast( timestamps, values, "randomForest" )
FROM inputStream;
```

Data Preprocessing: Before fitting data into sophisticated models, it is usually necessary to transform and preprocess the data. Striim supports a collection of data preprocessing methods including common approaches like standardization and normalization to make training less sensitive to the scale of features, power transformation(like square root, log) to stabilize volatile data, and seasonality decomposition like Loess or STL decomposition[3] to remove the trends and seasonality in time-series data, etc.

Automated Prediction: Striim has a plethora of machine learning models for clustering, classification, and regression. The models include linear and non-linear regression, SVM, gaussian process regression, random forest, Bayesian networks, deep learning models, etc. After preprocessing the data, one can fit the data into any of them to train a model. A common scenario is to predict the future values for real-time data. i.e., time series regression. In Striim, it first transforms time series regression to a standard regression progression problem with lagged variables, and then applies any regression model to it for prediction. Specifically, variable z_t is predicted by lagged variables $z_{t-1}, z_{t-2}, \ldots, z_{t-N}$, where N is the number of lagged variables.

Anomaly Detection: Striim detects anomalies based on its prediction. The underlying main idea is that anomalies tend to have

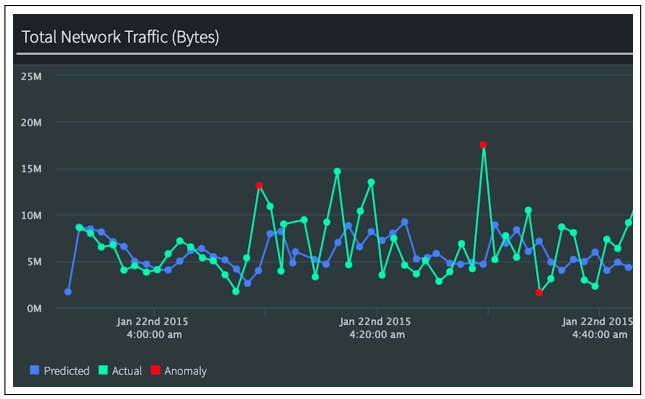

Figure 3: Striim Demonstration – Striim will first integrate the network traffic data, aggregate the total bytes of network traffic per minute for each IP address, analyze the actual data (green points) with machine learning to predict future traffic (blue points) and detect anomalies (red points).

large difference between actual value and predicted value. Some models can both predict the value and produce prediction intervals. e.g., Gaussian Process Regression produces a 95% prediction interval, which means the actual value should lie in the interval with 95% probability. For such models, Striim simply finds anomalies if actual values are out of the range. For other models which cannot produce prediction intervals, Striim uses a dynamic threshold to find anomalies if the percentage error between actual and predicted values is larger than the threshold. Striim assumes percentage errors follow Gaussian distribution and defines the dynamic threshold accordingly. e.g. we can define the threshold as $\mu + 2 * \sigma$, where μ is mean and σ is standard deviation of the percentage errors.

4 DEMONSTRATION

Our demonstration of Striim is meant to showcase its ability to collect, integrate, process, analyze and visualize streaming data to make real-time decisions. The demonstration is comprised of two applications. The first one is a very simple data collection application. As shown in Figure 2, it moves event data from an on-premise Oracle database to Google Cloud's Spanner, and transforms the data on the fly. It filters data whose user is Striim and value is not null. We demonstrate the ease of use of Striim: the users only need to drag and drop a source component , a SQL component, and a target component in the web UI, i.e., OracleReader, TransformSQL, and SpannerWriter shown in Figure 2. We will invite participants to rewrite the SQL-like query in TransformSQL component to transform, filter, enrich data using Striim's web interface. Simply deploy and run the application through user interface, the data will be moved from Oracle to Spanner and processed on the fly.

Another application is network traffic anomaly detection. We demonstrate how to use Striim to monitor the network traffic, predict the future network traffic, detect anomalies in network traffic, visualize the data and associated analysis, and send alerts in real time. This application is a little complex so that we pre-build the pipeline by ourselves. We will visualize the real-time network traffic data and detected anomalies, and invite the audience to see whether these detected anomalies are reasonable or not. They can also find anomalies by themselves and compare the results with Striim.

The dataset we use is called UNSW-NB15[5], which is a popular dataset for network traffic analysis. The dataset has about 2 million records, which are simulated tcpdump data from normal traffic and malicious traffic. In the application pipeline, Striim first reads data from Oracle databases, extracts features including IP address and port number of source and destination, transaction timestamp, and transaction bytes from data records. It then aggregates the total bytes of network traffic data for each IP address per minute. Next it predicts the future traffic data one-step ahead with machine learning models, detects anomalies if the actual network traffic is far from the predicted one, and writes results to Spanner. The models are updated periodically to guarantee the high prediction performance for continuous data streams. As shown in Figure 3, Striim detects some peaks and troughs as anomalies (red points). The intuition is that it is abnormal if the network traffic data suddenly increases or drops. The monitoring dashboard will be updated in real time. When a new anomaly is detected, it will show up immediately.

Striim supports a number of preprocessing and machine learning models as described in Section 3.2. In our pipeline, we apply log transformation to preprocess the data, fit it into random forest models, and use dynamic percentage error threshold to detect anomalies. We find such model selection has good performance according to our empirical evaluation. We will invite audience to choose different machine learning models, and compare the performance between them using the web UI. Audience can also choose different level of sensitivity to anomalies in Striim, e.g., low sensitivity means less anomalies will be detected.

Demo Takeaways: The goals of this demo are threefold. First, we seek to engage with the audience using a demonstration of the challenges of real-time integration and intelligence. Second, we will demonstrate Striim's ability to capture, integrate, process, analyze, visualize, and monitor over large streaming data in real time. Lastly, we hope to showcase the ease of use of Striim. One can simply drag and drop the components and write SQL-like queries using Striim's web interface to develop a streaming application.

5 CONCLUSION

We presented the design and architecture of Striim, a modern end-to-end streaming platform for real-time integration and intelligence. We described various features of the Striim that enables it to capture real-time data, integrate high-volume, high-velocity data streams and analyze it with sophisticated machine learning models in real time. We also introduced our demo plan and the applications, i.e., data collection and network traffic anomalies detection with Striim.

REFERENCES

[1] Striim. https://www.striim.com.
[2] P. Carbone, S. Ewen, S. Haridi, A. Katsifodimos, V. Markl, and K. Tzoumas. Apache flink: Stream and batch processing in a single engine. *IEEE Data Eng. Bull.*, 2015.
[3] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 1990.
[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD*, 2009.
[5] N. Moustafa and J. Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems. *MilCIS*, 2015.
[6] A. Pareek, B. Khaladkar, R. Sen, B. Onat, V. Nadimpalli, M. Agarwal, and N. Keene. Striim: A streaming analytics platform for real-time business decisions. *BIRTE*, 2017.
[7] A. Pareek, B. Khaladkar, R. Sen, B. Onat, V. Nadimpalli, and M. Lakshminarayanan. Real-time etl in striim. *BIRTE*, 2018.